Large-Scale Short-Term Planning in Chemical Batch Production

Norbert Trautmann
Department of Business Administration, University of Bern, 3012 Bern, Switzerland,
norbert.trautmann@ifm.unibe.ch

Christoph Schwindt
Institute of Management and Economics, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany,
christoph.schwindt@tu-clausthal.de

In the chemical industry, final products arise from chemical and physical transformations of materials in processing units. We consider the case of batch production mode, where the total requirements for intermediate and final products are divided into individual batches. To produce a batch, the inputs are first loaded into a processing unit, then a transformation process is executed, and finally the output is unloaded from the processing unit. In general, storage facilities of limited capacity are available for stocking raw materials, intermediates, and final products.

We present a novel cyclic approach to solving large-scale instances of the minimum-makespan production scheduling problem. This problem can be decomposed into a batching and a batch scheduling problem. The basic idea of the cyclic approach consists in reducing the size of the batch scheduling problem by computing a cyclic sub-schedule, which needs to be executed several times. Using a mixed-integer nonlinear programming formulation of the batching problem one can compute the set of batches of one cycle and the number of cycles needed to satisfy the primary requirements. The sub-schedule is then obtained by scheduling the batches on the processing units subject to material-availability and storage-capacity constraints. In an experimental performance analysis, we applied this cyclic approach to a set of 70 test instances. For each instance, we obtained a better feasible solution within much less CPU time than a state-of-the-art method from the literature.

Keywords: Applications, Production Scheduling, Process Scheduling, Large Scale Scheduling

1 Planning problem

Short-term planning of batch production in the chemical industry deals with the detailed allocation of the production resources of a single plant over time to the processing of given primary requirements for final products. Batch production is typically used either for technological reasons or for the case of multiple products processed on multi-purpose equipment. In Subsection 1.1 we review the particular characteristics of batch production on multi-product production plants. In Subsection 1.2 we state the planning problem. In Subsection 1.3 we introduce a practical example of a chemical production plant that has been provided by Kallrath (2002).

1.1 Batch production

In general, a multi-product plant consists of multi-purpose processing units (e.g., heaters, filters, and reactors) and storage facilities (e.g., tanks, silos, and a cooling house). The final products are produced by performing a sequence of transformations, which are also called tasks. For executing a task, several alternative processing units may be available. In this case, the duration of the task may depend on the processing unit used. In a multi-purpose processing unit, several processes can be performed, but only one at a time. Between consecutive executions of different tasks in a processing unit, a cleaning with sequence-dependent duration may be necessary.

Each task consumes and produces one or several products, where the input or output proportions are either fixed or variable within prescribed bounds. Some intermediates are perishable and must be consumed immediately after production. Material flows can be linear, divergent, convergent, or general (including the case of recycling flows).

The minimum and maximum filling levels of the processing unit used give rise to a lower and an upper bound on the batch size. This is the reason for executing a task several times to fulfill the primary requirements. Note that in chemical batch production, the duration of a task is independent of the batch size. In the following, the execution of a task will be called an operation.

Each product family requires a specific configuration of the plant. During a re-configuration of the plant, no operation can be processed. Thus, the objective of makespan minimization is particularly important in order to ensure high resource utilization and short customer lead times.

1.2 Short-term planning problem

The planning problem can be stated as follows. Given primary requirements for the final products, we must determine (a) the batch size, the input and the output proportions, and the number of executions for each task; (b) an assignment of the corresponding operations to the processing units; and (c) start times of the operations such that

- the given primary requirements for final products are satisfied,
- the prescribed intervals for the batch sizes and the input and output proportions are observed,
- no processing unit processes more than one operation at a time,
- the processing units are cleaned between consecutive operations,
- a sufficient amount of each input product is available at the start of each operation,
- sufficient storage space for output products is available at the completion of each operation,
- all perishable intermediates are consumed immediately after production, and
- the makespan is minimized.

1.3 Sample production process

In this subsection we describe the chemical batch production process of the case study presented by Kallrath (2002) and based on an existing plant. For the representation we use a modification of the state-task network (STN) concept introduced by Kondili et al. (1993). An STN is a directed graph which includes three types of elements:

1. *State nodes* represent the raw materials, intermediates, and final products. They are drawn as ellipses labeled with the respective state number and the initial and maximum stocks of the corresponding product. Some of the intermediate products cannot be stocked, which is indicated by the label "ns" (no stock). The value ∞ for the initial or maximum stock means that there is respectively sufficient initial stock or storage capacity.
2. *Task nodes* refer to the chemical or physical transformations of materials from one or more input states into one or more output states. Task nodes are represented by rectangles indicating the task number, the processing units in which the task can be executed, and the corresponding processing and cleaning times.

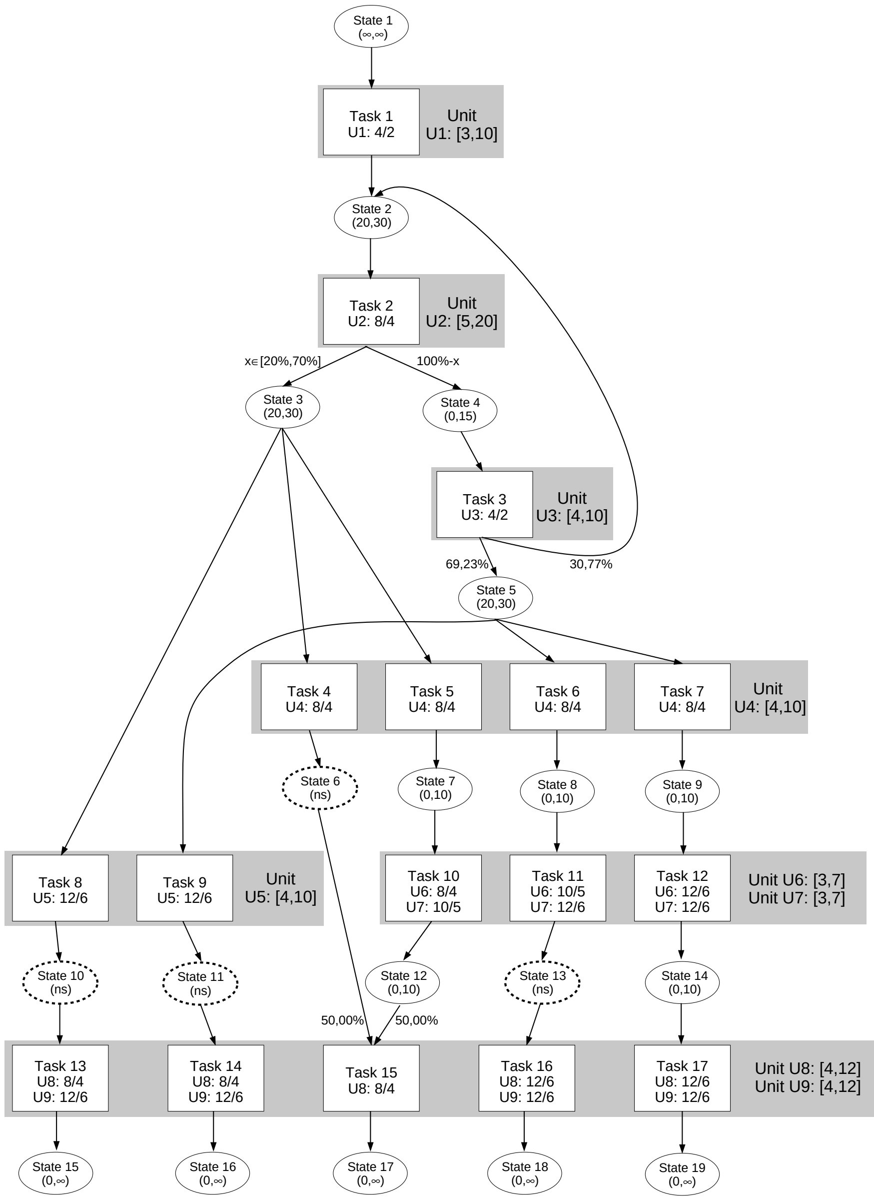

Figure 1: State-task network of the chemical batch production process

3. *Arcs* indicate the flow of material. If more than one input product is consumed or more than one output product is produced, the possible values of the input or output proportions are shown on the arcs.

Figure 1 shows the STN for the batch production process under study with 19 products, 17 tasks, and 9 processing units. The shaded areas group the tasks that can be processed in the same units. Alternative processing units are available for executing tasks 10 to 14, 16, and 17. In order to guarantee product purity, each processing unit must be cleaned before proceeding to an operation at a higher task index (tasks are numbered according to increasing quality requirements). The time needed for cleaning a processing unit equals one half of the processing time of the preceding operation.

2 Related literature

In the context of supply chain management, the primary requirements to be produced in the production network are determined on the mid-term campaign planning level. Campaign planning aims at using the procurement, production, storage, and transportation facilities in the supply chain efficiently by synchronizing the respective material flows. For mixed-integer linear programming models for campaign planning, we refer to Grunow et al. (2002) and Timpe and Kallrath (2000).

The short-term planning problem described in Subsection 1.2 has been widely discussed in the chemical engineering literature. An overview of state-of-the-art models and methods can be found in the survey papers of Floudas and Lin (2004), Burkard and Hatzl (2005), and Méndez et al. (2006). Roughly speaking, monolithic approaches (cf. Subsection 2.1) and decomposition approaches (cf. Subsection 2.2) can be distinguished.

2.1 Monolithic approaches

The monolithic solution approaches address the short-term planning problem as a whole, starting from a mixed-integer linear programming formulation. The time horizon is divided into a given number of time periods. In so-called time-indexed formulations (see e.g., Kondili et al. 1993), the period length is fixed. In contrast, in so-called continuous-time formulations (see e.g., Ierapetritou and Floudas 1998 or Castro et al. 2001), the period length is chosen implicitly during the solution of the mixed-integer linear program.

The main disadvantage of all these monolithic approaches is that the CPU time required for solving real-world problems tends to be prohibitively long (cf. Maravelias and Grossmann 2004). To overcome this difficulty, Shah et al. (1993), Blömer and Günther (2000), and others have developed different heuristics that aim at reducing the number of variables. Nevertheless, the computational burden for solving real-world problems with more than 50 operations is still very high.

2.2 Decomposition approaches

Promising alternative approaches are based on decomposing the short-term planning problem into interdependent subproblems. Decomposition methods have for example been proposed by Brucker and Hurink (2000), Neumann et al. (2002), and Maravelias and Grossmann (2004).

Brucker and Hurink (2000) did not consider all the constraints mentioned in Subsection 1.1. In particular, they assumed that the capacity of the storage facilities is unlimited, and that each task can be executed in one dedicated processing unit. The authors have devised a constructive algorithm for computing the numbers and the sizes of the batches, and a tabu search procedure for scheduling the batches on the processing units.

Maravelias and Grossmann (2004) proposed computing the number of batches by solving the LP relaxation of a monolithic continuous-time formulation of the short-term planning problem. The batch sizes and the start times of the operations were then determined by a branch-and-bound algorithm that uses constraint-propagation techniques.

The solution approach proposed in the present paper is based on a hierarchical decomposition into a batching and a batch-scheduling problem presented in Neumann et al. (2002). The solution of the batching problem provides the numbers and the sizes of all batches for the intermediate and final products needed to satisfy the primary requirements. The batch scheduling problem consists in allocating the processing units, intermediates, and storage facilities over time to the processing of the operations arising from the batching step. In Neumann et al. (2002), the batching problem was formulated as a mixed-integer nonlinear program which is of moderate size and can be solved using standard mathematical programming software. Neumann et al. (2002) and Schwindt and Trautmann (2004) have developed a truncated branch-and-bound method and a priority-rule-based method, respectively, for solving the batch scheduling problem. Within a reasonable amount of computation time, good feasible solutions to problem instances with up to 100 operations can be computed with both methods. Gentner et al. (2004) proposed a decomposition of the batch scheduling problem which partitions the set of all batches into a sequence of subsets. The assignment of the batches to the individual subsets is determined stepwise by solving a binary linear program in each iteration. Gentner et al. (2004) and Gentner (2005) computed feasible solutions to batch scheduling instances with up to 3000 operations. However, for such large-scale instances, this method requires several hours of CPU time.

3 Cyclic solution approach

In this section we present a cyclic approach to the short-term planning problem, which is based on the decomposition principle introduced by Neumann et al. (2002). A preliminary version of our approach can be found in Schwindt and Trautmann (2006).

Our method consists of the three phases of cyclic batching (Subsection 3.1), cyclic batch scheduling (Subsection 3.2), and concatenation (Subsection 3.3). Each of these phases is performed only once. Moreover, we limit the size of the cyclic batch scheduling problem to be solved. In total, a relatively short CPU time is required, and we are able to efficiently cope with problem instances including thousands of operations.

3.1 Cyclic batching

In the cyclic batching phase, we determine the set of operations (together with their respective batch sizes and input and output proportions) belonging to one cycle, and the number of cycles needed to satisfy the given primary requirements. In doing so we must take into account the prescribed bounds on the batch sizes, the prescribed bounds on the input and output proportions, and the initial inventory levels. Moreover, in order to keep the scheduling problem tractable, we impose an upper bound on the total number of operations belonging to one cycle. To obtain a cyclic solution allowing for executing the same sub-schedule an arbitrary number of times, the amount of any intermediate produced within one cycle must be equal to the amount consumed. This cyclic batching problem can be formulated as a mixed-integer nonlinear program of moderate size (cf. Schwindt and Trautmann 2006), and locally optimal solutions can be determined using standard software.

3.2 Cyclic batch scheduling

In the cyclic batch scheduling phase, we compute a sub-schedule by allocating the processing units and storage facilities over time to the processing of the operations belonging to one cycle such that the makespan is minimized. An appropriate sub-schedule can be determined using the truncated branch-and-bound method or the priority-rule-based method proposed by Neumann et al. (2002) and Schwindt and Trautmann (2004), respectively. An improved version of the latter procedure is presented in Fink and Schwindt (2007).

The main principle of the priority-rule-based method consists in scheduling the operations one after another on the processing units in such a way that the material-availability constraints are observed. The storage-capacity constraints are taken into account in a second scheduling pass (Schwindt and Trautmann's method) or by appropriately delaying producing operations via an unscheduling procedure (Fink and Schwindt's method).

3.3 Concatenation

In the concatenation step, we generate a complete production schedule as follows. The computed sub-schedule for executing the operations of one cycle defines a partial ordering among those operations. We represent this ordering by precedence relationships between the operations. Moreover, the completion time of the last operation that is processed in a processing unit defines a release date for the changeover to the first operation in that unit in the next execution of the sub-schedule. Analogously, the last change in the inventory level of an intermediate gives rise to a release date for the first operation that subsequently produces or consumes that intermediate.

The start and completion times for the operations in the first cycle equal those of the sub-schedule computed in the cyclic batch scheduling phase. For computing the start and completion times of the operations in the next cycle, we solve a temporal scheduling problem, which consists in computing an earliest schedule for those operations subject to the precedence relationships between and the release dates for the operations. This temporal scheduling problem represents a longest path problem and can be solved efficiently by standard network flow algorithms (see Ahuja et al., 1993). Thus, the concatenation of the cyclic sub-schedules forming the complete production schedule can be performed in polynomial time.

4 Performance analysis

We compared our cyclic approach to the decomposition method devised by Gentner et al. (2004). For our analysis, we used a test set proposed by Gentner (2005), which consists of 70 instances generated by varying the primary requirements for the final products in the example presented in Subsection 1.3. For each instance we computed an approximate solution to the cyclic batching problem using Frontline Systems' Solver package. The sub-schedules were generated with a randomized multi-pass version of Schwindt and Fink's priority-rule-based method (2007). We performed the tests on an 3.4 GHz Pentium IV PC. The results for the method of Gentner et al. have been reported in Gentner (2005) and refer to a 1.4 GHz Pentium IV PC.

The results obtained for the 70 problem instances are shown in Table 1, where "$C_{\max}$" stands for the best makespan found, "t_{cpu}" is the CPU time in seconds, and "# op.'s" designates the number of operations in the complete production schedule. For each problem instance the new method was able to find a markedly better solution. Especially for large-scale problem instances, the required CPU time was significantly smaller than the time needed by the method of Gentner et al. Having prescribed an upper bound of $\bar{\varepsilon} = 150$ batches, between 14 and 583 seconds were required for solving the cyclic batching problem. The priority-rule based method was stopped after 60 seconds of CPU time. The concatenation required less than one second of CPU time.

5 Conclusions

In this paper we have presented a cyclic approach to short-term planning in chemical batch production. Our method is based on the decomposition of the short-term planning problem into a batching level providing the set of operations to be executed and a batch scheduling level that schedules the operations on the processing units subject to material-availability and storage-capacity constraints. The main idea of our cyclic approach consists in formulating the batching problem as a cyclic model where the given primary requirements are produced through the repetitive execution of the same set of operations. In this way we ensure that the resulting batch scheduling problem can be solved within a reasonable amount of computation time by computing a subschedule for the operations of one cycle and concatenating the number of cycles needed to meet the primary requirements.

Table 1: Computational results

Instance	Gentner (2005) $C_{\max}$	Gentner (2005) t_{cpu}	This paper # op.'s	This paper $C_{\max}$	This paper t_{cpu}
WeKa0_0	178	18	88	128	116
WeKa0_1	352	38	176	252	113
WeKa0_2	474	53	264	376	118
WeKa0_3	612	120	352	500	119
WeKa0_4	738	209	440	624	115
WeKa0_5	906	178	528	748	122
WeKa0_6	1046	215	616	872	119
WeKa0_7	1199	323	704	996	121
WeKa0_8	1334	281	792	1120	117
WeKa0_9	1548	399	880	1244	128
WeKa0_10	1740	431	968	1368	100
WeKa0_15	2123	644	1408	1988	97
WeKa0_20	2899	1500	1848	2608	97
WeKa0_30	4416	5235	2728	3884	77
WeKa19_0	238	19	105	166	80
WeKa19_1	436	165	210	316	81
WeKa19_2	618	59	315	466	79
WeKa19_3	818	97	420	616	80
WeKa19_4	1004	179	525	766	81
WeKa19_5	1184	232	630	916	80
WeKa19_6	1384	330	735	1066	83
WeKa19_7	1570	474	840	1216	81
WeKa19_8	1806	442	945	1366	81
WeKa19_9	1946	568	1050	1516	80
WeKa19_10	2135	570	1155	1666	83
WeKa19_15	2848	1322	1680	2416	79
WeKa19_20	3811	1911	2205	3166	78
WeKa19_30	5896	6610	3255	4666	76
WeKa20_0	168	34	89	138	86
WeKa20_1	336	50	178	264	87
WeKa20_2	590	72	267	390	90
WeKa20_3	750	76	356	516	100
WeKa20_4	896	93	445	642	98
WeKa20_5	990	126	534	768	100
WeKa20_6	1138	184	623	894	95
WeKa20_7	1294	215	712	1020	95
WeKa20_8	1547	200	801	1146	96
WeKa20_9	1816	327	890	1272	94
WeKa20_10	1920	448	979	1398	94
WeKa20_15	2386	421	1424	2028	96
WeKa20_20	3604	969	1869	2658	96
WeKa20_30	5194	3255	2759	3918	75
WeKa21_0	210	17	98	144	103
WeKa21_1	382	127	196	284	100
WeKa21_2	555	67	294	424	95
WeKa21_3	728	97	392	564	91
WeKa21_4	868	152	490	704	86
WeKa21_5	1082	226	588	844	86
WeKa21_6	1224	250	686	984	83
WeKa21_7	1420	240	784	1124	82
WeKa21_8	1554	291	882	1264	85
WeKa21_9	1701	475	980	1404	85
WeKa21_10	1916	469	1078	1544	82
WeKa21_15	2545	771	1568	2244	81
WeKa21_20	3398	1415	2058	2944	82
WeKa21_30	5091	5957	3038	4344	89
WeKa22_0	190	192	102	152	327
WeKa22_1	376	85	204	290	644
WeKa22_2	558	102	306	428	298
WeKa22_3	722	120	408	566	155
WeKa22_4	930	249	510	704	239
WeKa22_5	1024	239	612	842	324
WeKa22_6	1298	255	714	980	270
WeKa22_7	1488	341	816	1118	150
WeKa22_8	1520	439	918	1256	276
WeKa22_9	1779	427	1020	1394	149
WeKa22_10	1786	647	1122	1532	221
WeKa22_15	2586	704	1632	2222	171
WeKa22_20	3172	1598	2142	2912	206
WeKa22_30	5375	7563	3162	4292	271

The performance of the new method has been tested on a test set involving instances with several thousands of operations.

An important area of our future research will be the adaptation of the cyclic approach to continuous process scheduling problems where tasks are executed at constant production rates. In addition, we are developing predictive-reactive methods for the short-term planning of chemical production plants when processing times, resource availabilities, or production yields are subject to uncertainty. The different short-term planning methods will be integrated with decision models for mid-term multi-site campaign planning in the chemical industry.

Acknowledgements. This research has been supported by the Deutsche Forschungsgemeinschaft under Grant Schw1178/1.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin (1993), *Network Flows*, Prentice Hall, Englewood Cliffs.

[2] F. Blömer, H.O. Günther (2000), LP-based heuristics for scheduling chemical batch processes. *Int. J. Prod. Res.* **38**, 1029 – 1051.

[3] P. Brucker, J. Hurink (2000), Solving a chemical batch scheduling problem by local search, *Annals of Oper. Res.* **96**, 17 – 36.

[4] R.E. Burkard, J. Hatzl (2005), Review, extensions and computational comparison of MILP formulations for scheduling of batch processes, *Comp. Chem. Eng.* **29**, 1752 – 1769.

[5] P. Castro, A.P. Barbosa-Póvoa, H. Matos (2001), An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants, *Ind. Eng. Chem. Res.* **40**, 2059 – 2068.

[6] R. Fink, C. Schwindt (2007), A priority-rule based method for predictive-reactive batch production scheduling in the process industries. *Report PLC–3*, Institute of Management and Economics, Clausthal University of Technology.

[7] C.A. Floudas, X. Lin (2004), Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review, *Comp. Chem. Eng.* **28**, 2109 – 2129.

[8] K. Gentner (2005), *Dekompositionsverfahren für die ressourcenbeschränkte Projektplanung*, Shaker, Aachen.

[9] K. Gentner, K. Neumann, C. Schwindt, N. Trautmann (2004), Batch production scheduling in the process industries. In: J.Y.T. Leung (ed.), *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, CRC Press, Boca Raton, Chapter 48, 1 – 21.

[10] M. Grunow, H.O. Günther, M. Lehmann (2002), Campaign planning for multi-stage batch processes in the chemical industry, *OR Spectrum* **24**, 281 – 314.

[11] M.G. Ierapetritou, C.A. Floudas (1998), Effective continuous-time formulation for short-term scheduling: 1. Multipurpose batch processes, *Ind. Eng. Chem. Res.* **37**, 4341 – 4359.

[12] J. Kallrath (2002), Planning and scheduling in the process industry, *OR Spectrum* **24**, 219 – 250.

[13] E. Kondili, C.C. Pantelides, R.W.H. Sargent (1993), A general algorithm for short-term scheduling of batch operations — I. MILP Formulation, *Comp. Chem. Eng.* **17**, 211 – 227.

[14] C.T. Maravelias, I.E. Grossmann (2004), A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants, *Comp. Chem. Eng.* **28**, 1921 – 1949.

[15] C.A. Méndez, J. Cerdá, I.E. Grossmann, I. Harjunkoski, M. Fahl (2006), State-of-the-art review of optimization methods for short-term scheduling of batch processes, *Comp. Chem. Eng.* **30**, 913 – 946.

[16] K. Neumann, C. Schwindt, N. Trautmann (2002), Advanced production scheduling for batch plants in process industries, *OR Spectrum* **24**, 251 – 279.

[17] C. Schwindt, N. Trautmann (2004), A priority-rule based method for batch production scheduling in the process industries. In: D. Ahr, R. Fahrion, M. Oswald, G. Reinelt (eds.), *Operations Research Proceedings 2003*, Springer, Berlin, 111 – 118.

[18] C. Schwindt, N. Trautmann (2006), A cyclic approach to large-scale short-term planning of multi-purpose batch plants. In: M. Morlock, C. Schwindt, N. Trautmann, J. Zimmermann (eds.), *Perspectives on Operations Research*, Deutscher Universitäts-Verlag, Wiesbaden, 225 – 238.

[19] N. Shah, C.C. Pantelides, R.W.H. Sargent (1993), A general algorithm for short-term scheduling of batch operations — II. Computational Issues, *Comp. Chem. Eng.* **17**, 229 – 244.

[20] C.H. Timpe, J. Kallrath (2000), Optimal planning in large multi-site production networks, *Eur. J. Oper. Res.* **126**, 422 – 435.